artdaq Utilities - Feature #21226

DAQInterface should support multiple sub-configurations

10/24/2018 03:20 PM - John Freeman

Status:	Resolved	**Start date:**	10/24/2018
Priority:	Normal	**Due date:**	
Assignee:	John Freeman	**% Done:**	100%
Category:	artdaq-daqinterface	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Experiment:	-	**Co-Assignees:**	

Description

A request that was brought up during Monday's protoDUNE call and which we agreed was a good idea later that day in the weekly RSI meeting was that, rather than only supporting a monolithic configuration whose name refers to all FHiCL documents needed for a run - EventBuilders, various fragment generators, etc. - DAQInterface should also allow an experiment's users to label a subset of the FHiCL documents as its own configuration (or perhaps the right word is "subconfiguration"?). The idea behind this is that it would allow experts for a particular subsystem/fragment generator to modify their subconfiguration without accidentally reverting changes made by another set of experts which concerned their other subystem/fragment generator. A good first step in this direction would be to allow DAQInterface to accept multiple arguments during the config transition, i.e., something like this:

```
send_transition.sh config latest_eventbuilders latest_dataloggers latest_routingmaster subystem1_boardreader_fhicl subsystem2_boardreader_fhicl # ...etc
```

...and then DAQInterface could access the union of the FHiCL documents across these configurations when sending the init transition to artdaq processes.

History

#1 - 10/31/2018 09:51 AM - John Freeman

More info:

- It's possible that a FHiCL document of the same name could appear in multiple provided subconfigurations. One approach to handle this would be to have the FHiCL document in the subconfiguration which appears later in the argument list to clobber the document which appears earlier, and to issue a warning. However, I prefer the idea of DAQInterface simply throwing an exception, as this scenario most likely indicates that there's some confusion on the user's end that needs to be cleared up.

- This should be backwards compatible - i.e., DAQInterface users, if they wish to continue using a single labeled configuration, should be able to do so without ever realizing that the software was extended to handle multiple subconfigurations.

- A word on language. If you execute send_transition.sh config myconfig_boardreaders myconfig_eventbuilders then myconfig_boardreaders and myconfig_eventbuilders would each be referred to as subconfigurations. The configuration in this case would be called "myconfig_boardreaders myconfig_eventbuilders", and would appear as such in the metadata.txt file saved in the run record. Notice that in this approach, if users provide just one argument to the config transition, while technically that argument could be referred to as a subconfiguration, it could also technically be referred to as a configuration - and in this sense backwards compatibility is preserved not only in the sense of the way the code behaves, but also in the sense of how things get referred to.

#2 - 10/31/2018 11:04 AM - Kurt Biery

John, thanks for the additional info, but it would be great to understand more about the design of what is being changed. The additional info that you added seems to assume that the reader already knows the design and seems to dive down into implementation details.

Some questions that I have about the design:

- is the goal of this change to provide the option to move existing FHiCL documents into separate subconfigurations, without changing the structure or contents of the documents?
- can each subconfiguration have a "common_code" area?
- can a document in one subconfiguration include a snippet from the common_code area in another subconfiguration?
- how will snippets like the routing_master one that we are using on protoDUNE be handled, since it is included by BR configs in several detector subsystems?
- will bookkeeping be done on the files after they are merged into the full set, or before they are merged?

#3 - 10/31/2018 11:50 AM - John Freeman

Just had a discussion with Kurt, mainly focused on the questions he poses above. To answer his questions point-by-point:

- Yes, the structure or contents can be unchanged.

- Not only can each subconfiguration have a common_code area, it can have an areas (subdirectories) with **any** name (common_code or otherwise) which can be added to the search path for #includes

- A document in one subconfiguration can include snippets from common_code areas (or any other areas) in other subconfigurations. In other words, the search path described in the previous point crosses all the subconfigurations. We both agreed that for starters, DAQInterface should demand unique filenames across all subconfigurations, which makes this requirement easy to satisfy (no need to decide what the "best" area is if an #included file is found in more than one location).

- Because snippets can be taken from anywhere across the subconfigurations, routing_master is easily handled. One could even imagine a subconfiguration which just contains the routing master's FHiCL document.

- Bookkeeping will be done after the files are merged, i.e., is unrelated to how many subconfigurations were needed to create the complete set of FHiCL documents.

One issue that came up is that because FHiCL documents are more decoupled from one another if they're spread across multiple subconfigurations than if they're in one subconfiguration (the latter being what we've effectively been doing all along up until this point), there's a greater risk of occurrences such as, say, boardreader FHiCL documents including logic to send info to a routing master but eventbuilder FHiCL documents not using a routing master. If this becomes a problem, we can discuss the possibility of having DAQInterface enforce invariants across FHiCL documents.

#4 - 11/02/2018 10:02 AM - John Freeman

- Status changed from New to Resolved

- % Done changed from 0 to 100

With commit 895754d36c83fa750f3faf692df23d8a6f8a47c5 on DAQInterface's develop branch (the result of a fast-forward merge of feature/use_subconfigurations), the following applies:

- When given a single argument for the config transition, DAQInterface behaves exactly as before (+)
- However, you can now provide multiple arguments to the config transition. E.g.,

  ```
  just_do_it.sh  $PWD/boot.txt 10 --config "group1_boardreaders group2_boardreaders non_boardreaders" --comp
  s "group1board group2board" 
  ```

  or

  ```
  send_transition.sh config group1_boardreaders group2_boardreaders non_boardreaders
  ```

  ...and when this happens, all paths associated with the subconfigurations "group1_boardreaders", "group2_boardreaders" and "non_boardreaders" get added to the FHiCL file search path - i.e., we take the combination of all subconfigurations to "build" the configuration. If a duplicate filename is found across the subconfigurations, DAQInterface will throw an exception. The configuration is referred to as the combination of subconfigurations, so, e.g., in the metadata.txt file you'd see:

  ```
  Config name: group1_boardreaders group2_boardreaders non_boardreaders
  ```

(+) With the exception that the saved metadata.txt file may look a bit different. This is because in the metadata.txt file it's made explicit that common_code, should it exist, gets picked up automatically when we're working with configurations on the local filesystem rather than in the database. So, e.g., while it **remains** the case that common_code gets picked up automatically, now in the metadata you'd see something like "Config name: demo common_code" rather than just "Config name: demo"